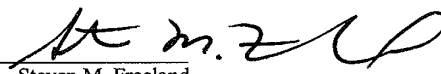


IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appln. No.: 09/649,215	<p align="center"><u>CERTIFICATE OF TRANSMISSION/MAILING</u></p> <p>I hereby certify that this correspondence is being facsimile transmitted to the USPTO, transmitted via the Office electronic filing system, or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date shown below:</p> <p>9/7/07 </p> <p>Date Steven M. Freeland Registration No. 42,555</p>
Applicants Lamkin et al.	
Filed: August 28, 2000	
Title: SOFTWARE ENGINE FOR COMBINING VIDEO OR AUDIO CONTENT WITH PROGRAMMATIC CONTENT	
Examiner: VU, Tuan A.	
Art Unit: 2193	
Customer No.: 22242	
Confirm. No.: 7416	

SUBSTITUTE APPEAL BRIEF

Mail Stop: APPEAL BRIEF - PATENT
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, Virginia 22313-1450

Dear Sir:

Appellants submit this appeal brief under 37 C.F.R. § 41.37 appealing the final rejection of Claims 1-12 in the final office action mailed September 28, 2006.

(1) Real Party in Interest

The real party in interest is Sonic Solutions.

(2) Related Appeals and Interferences

No related appeals or interferences are known to Appellants.

(3) Status of Claims

Claims 1-10 were submitted for examination in the application filed on March 28, 2003.

Claims 11 and 12 were added during prosecution.

Claims 1, 8, 9 and 10 were amended.

Claims 1-12 remain pending.

Claims 1-12 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,909,551 (Tahara).

Claims 1-12 are appealed.

(4) Status of Amendments

No amendments have been filed subsequent to the final rejection mailed September 28, 2006.

(5) Summary of Claimed Subject Matter

Independent Claims Subject Matter Map

generating authoring output comprising a definition for a variable, and further comprising a representation of the video/audio content, the representation of the video/audio content defining how the video/audio content is to be displayed	Claim 1
selecting a source file (in some instances through code), the source file comprising the variable	Claims 1 and 10
create a copy of the source file	Claim 10
searching (in some instances through code) an instance of the source file for the variable, and replacing the variable with the definition for the variable within that instance of the source file (or the copy of the source file when relevant)	Claims 1, 8, 9 and 10
generating (in some instances through code) programmatic content in response to the searching	Claims 1, 8, 9 and 10
generating an image (in some instances through code) as a function of the programmatic content and the representation of the video/audio content	Claims 1, 8, 9 and 10
the representation of the audio/video content defining how the video/audio content is to be displayed	Claims 8, 9 and 10
combining (in some instances format and combine) the image with the video/audio content (in some instances through code)	Claims 1, 8, 9 and 10

The claimed subject matter is directed to combining multimedia content with programmatic content. Further, in some embodiments the claimed subject matter provide for the authoring of video and/or audio content, and programmatic content for storage on or transmission through a medium.¹ Some embodiments further provide software engines for authoring video and/or audio content, and programmatic content for storage on or transmission through a medium.²

More specifically, some embodiments as recited in Claim 1 are directed to methods for combining video/audio content with programmatic content, other embodiments as recited in Claim 8 provides a system for combining video/audio content with programmatic content.³ Other embodiments as recited in Claim 9 provide computer program products stored on a computer-readable medium for use in combining video/audio content with programmatic content, and/or software system as recited in Claim 10 for combining video/audio content with programmatic content.⁴ Additionally or alternatively, some embodiments provide software systems stored on a computer readable medium.⁵ FIGS. 2-3 from the application appears below for the convenience of the reader showing examples of a system for combining video/audio content and a process to implement a combination of video/audio content according to some embodiments:

1 See at least App. FIGS. 1-3; pg. 2, lns. 17-24; pg. 4, ln. 7 – pg. 5, ln. 5; pg. 8, ln. 16 – pg. 19, ln. 24.

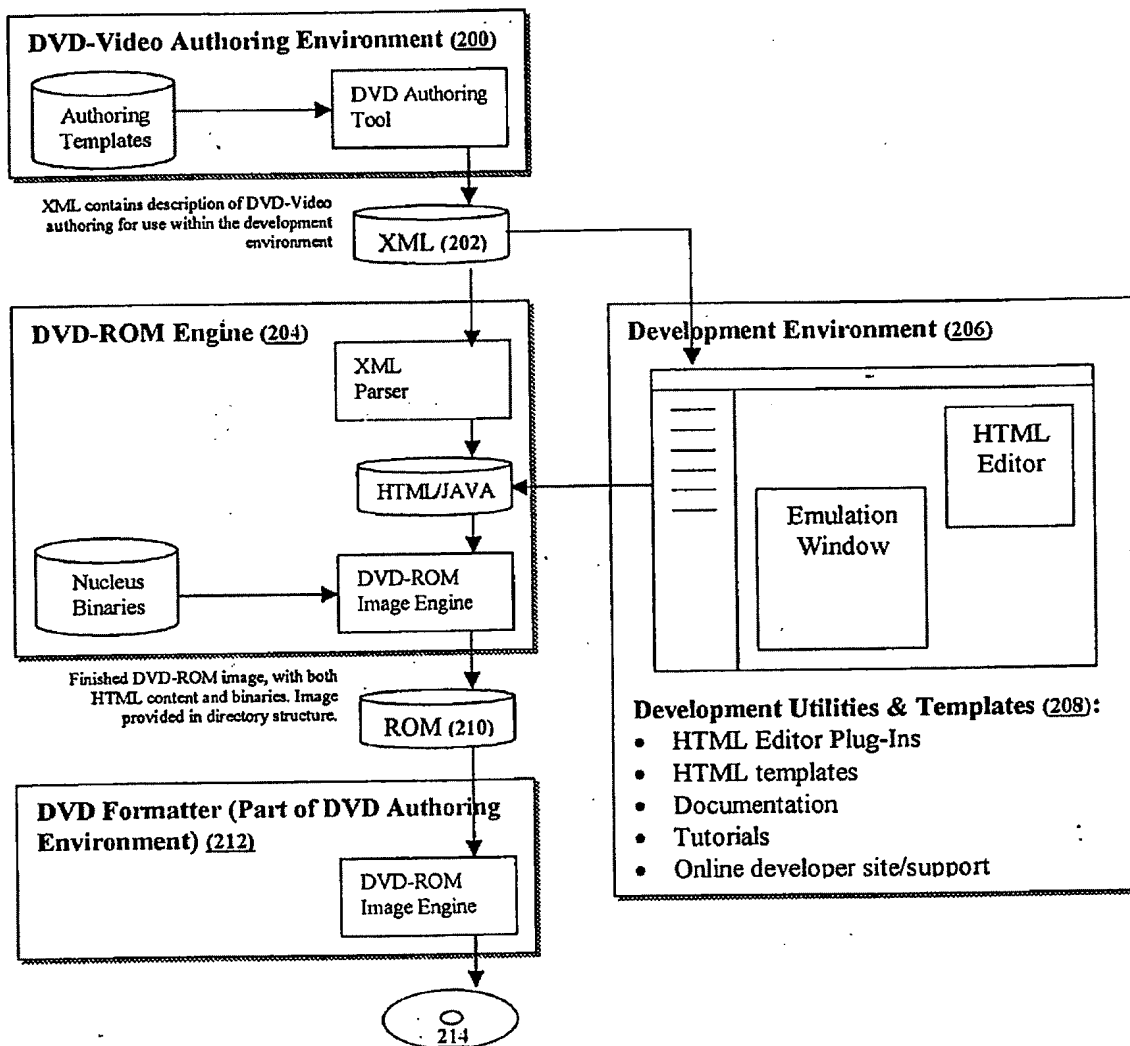
2 See at least App., FIGS. 1-3; pg. 2, lns. 17-24; pg. 4, ln. 7 – pg. 5, ln. 5; pg. 8, ln. 16 – pg. 19, ln. 24.

3 See at least App., FIGS. 1-3; pg. 4, lns. 7-26.

4 See at least App., FIGS. 1-3; pg. 4, ln. 7 – pg. 5, ln. 14.

5 See at least App., FIGS. 1-3; pg. 4, ln. 7 – pg. 5, ln. 14.

FIG. 2



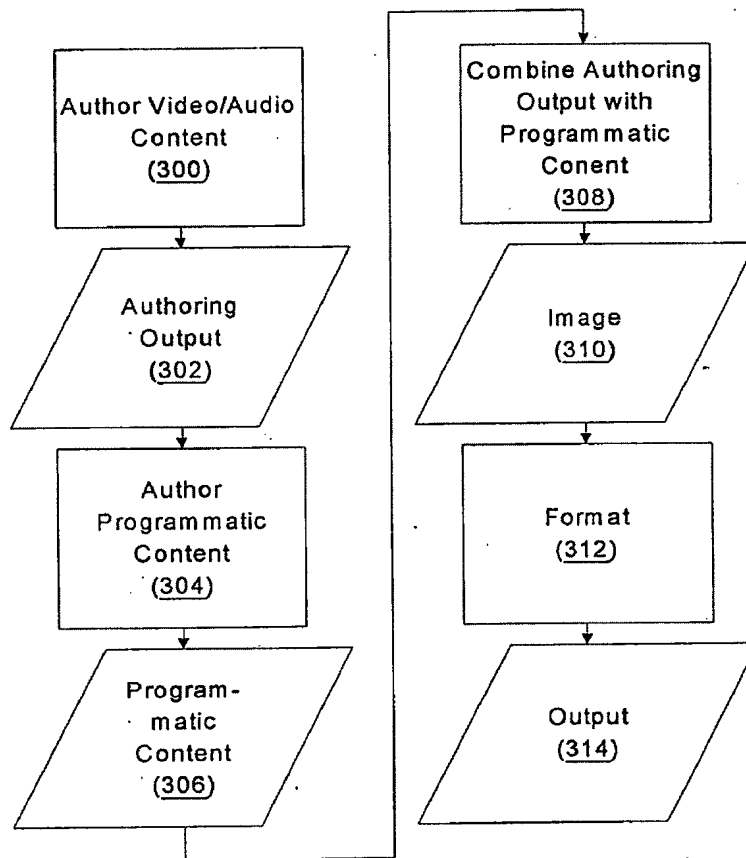


FIG. 3

Some embodiments combine video/audio content (100, 102, 200, 202, 300, 302) with programmatic content (104, 208, 304).⁶ Video/audio content is generated, and based at least in part on that video/audio content an authoring output (102, 202, 302) is further generated⁷, which in some embodiments comprises at least one definition for at least one variable (102, 202, 302).⁸ One or more source files can be generated and/or selected with each source file containing at least one variable.⁹ In some embodiments the source files are JavaScript and/or HTML source files.¹⁰

⁶ See at least App. FIGS. 1-3; pg. 4, lns. 7-30; pg. 6, ln. 14 – pg. 19, ln. 24; pg. 8, lns. 16-22; pg. 23, ln. 25 – pg. 24, ln. 14.

⁷ See for example, FIGS. 1-3; pg. 6, ln. 21 – pg. 7, ln. 2; pg. 9, lns. 7-20; pg. 12, lns. 7-14; pg. 23, ln. 27 – pg. 24, ln. 1.

⁸ See at least App., FIG. 2; pg. 6, ln. 29 – pg. 7, ln. 2; pg. 23, lns. 27-28

⁹ See at least App., FIGS. 1-3; pg. 7, lns. 2-11 and 25-30; pg. 9, lns. 14-20; pg. 12, lns. 7-14; pg. 15, ln. 25 – pg. 22, ln. 5; pg. 22, lns. 21-22; pg. 24, lns. 4-8.

¹⁰ Id.

With regard to at least Claim 1, an instance of the source file is then be searched to locate the variable that is defined within the authoring output (104, 106, 204, 206, 304, 306, 308).¹¹ The searching of the instance of the source file can occur at run time and/or at a build time.¹² In some embodiments the searching of the source file may occur in response to a software engine executed on a browser, and/or the searching can be implemented in response to the insertion of a DVD into hardware.¹³

The variable can be replaced within the source files with the definition for that variable contained within the authoring output (104, 106, 204, 206, 304, 306, 308) such that the instance of the source file contains the definition for the variable¹⁴, and to generate programmatic content (104, 106, 204, 206, 304, 306).¹⁵ In one or more embodiments, after the system generates or selects a source file, the system creates a copy of the source file and searches that instance of the source file for the variable, and replaces the variable with the definition for that variable within that copy of the source file.¹⁶ Further, an image (108, 210, 308, 310) is authored as a function of the programmatic content and the representation of the video/audio content and combines the image with the video/audio content (104, 106, 108, 204, 206, 210, 308, 310).¹⁷ In some embodiments the resulting combination is then stored onto a storage medium (110, 112, 212, 214, 312, 314).¹⁸ Additionally or alternatively, the image can be transmitted through a transmission medium.¹⁹

With regard to at least Claim 9, further embodiments are directed to computer program products stored on a computer-readable medium for use in combining video/audio content with programmatic content.²⁰ The computer program products, in some embodiment, parse a source file searching for a variable and replace the variable with a definition for the

11 See at least App., FIGS. 1-3, pg. 7, lns. 2-12, pg. 15, ln. 25 – pg. 22, ln. 5, pg. 23, ln. 29 – pg. 24, ln. 10.

12 See at least App. pg. 13, lns. 3-16; pg. 13, ln. 17 – pg. 15, ln. 7.

13 See at least, App. pg. 9, lns. 21-28; pg. 10, lns. 26-30

14 See at least App., FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 10-15, 19-24 and 29 - pg. 5, ln. 1; pg. 5 lns. 6-10; pg. 6, ln. 24 - pg. 8, ln. 22; pg. 9, lns. 13-20; pg. 12, ln. 7 - pg. 13, ln. 30; pg. 15, ln. 8 - pg. 22, ln. 5; pg. 23, ln. 28 - pg. 24, ln. 8.

15 See at least App., FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 10-15, 22-24 and 29-30; pg. 5, lns. 2-5 and 10-12; pg. 7, lns. 5-11; pg. 7, ln. 22 - pg. 8, ln. 22; pg. 11, ln. 6 – pg. 12, ln. 6; pg. 13, lns. 17-28; pg. 15, lns. 1-7 and lns. 19-24; pg. 19, lns. 16-21; pg. 22, ln. 6 – pg. 23, ln. 16; pg. 24, lns. 1-10.

16 See at least App.; pg. 19, lns. 1-4.

17 See at least App. FIGS. 1-3; pg. 4, ln. 16-26; pg. 5, lns. 3-14; pg. 7, lns. 5-11; pg. 8, lns. 6-8; pg. 11, ln. 24 - pg. 12, ln. 6; pg. 24, lns. 1-14.

18 For example, App., FIGS. 1-3; pg. 6, lns. 5-11; pg. 8, lns. 6-22; pg. 9, lns. 2-6; pg. 23, lines 17-24; pg. 24, lns. 12-14.

19 Id.

20 See for example, App., FIGS. 1-3; pg. 2, lns. 18-33; pg. 23, ln. 25 – pg. 24, ln. 14.

variable within the source file (100, 102, 106, 200, 202, 204, 300, 302).²¹ In some implementations, before searching the source file for the variable and replacing the variable with the definition for the variable the computer program products make a copy of the source file, and search the copy of the source file and replaces the variable within that copy of the source file.²² The computer program products can include further code that generates programmatic content in response to the searching (104, 105, 204, 206, 304, 306).²³ Further, in one or more embodiments, code to generate an image as a function of the programmatic content and a representation of the audio/video content is also included in the computer program products where in some embodiments the representation of the audio/video content defines how the video/audio content is to be displayed (104, 108, 204, 210, 308, 310).²⁴ Further, in some embodiments, code is provided to format and combine the image with the video/audio content (110, 112, 212, 214, 312, 314).²⁵

With regard to at least Claim 10, still other embodiments are directed to software systems stored on a computer readable medium.²⁶ The software systems, in some implementations, select a source file that includes a variable and make a copy of the source file.²⁷ The source file is searched for a variable and the variable within the copy of the source file is replaced with a definition for the variable (100, 102, 106, 200, 202, 204, 300, 302).²⁸ The software systems can include further code that generates programmatic content in response to the searching (104, 105, 204, 206, 304, 306).²⁹ Further, in one or more

21 See at least App., FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 10-15, 19-24 and 29 - pg. 5, ln. 1; pg. 5 lns. 6-10; pg. 6, ln. 24 - pg. 8, ln. 22; pg. 9, lns. 13-20; pg. 12, ln. 7 - pg. 13, ln. 30; pg. 15, ln. 8 - pg. 22, ln. 5; pg. 23, ln. 28 - pg. 24, ln. 8.

22 See at least App.; pg. 19, lns. 1-4.

23 See at least App., FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 10-15, 22-24 and 29-30; pg. 5, lns. 2-5 and 10-12; pg. 7, lns. 5-11; pg. 7, ln. 22 - pg. 8, ln. 22; pg. 11, ln. 6 - pg. 12, ln. 6; pg. 13, lns. 17-28; pg. 15, lns. 1-7 and lns. 19-24; pg. 19, lns. 16-21; pg. 22, ln. 6 - pg. 23, ln. 16; pg. 24, lns. 1-10).

24 See for example App. FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 16-18 and 24-26; pg. 5, lns. 3-5 and 12-14; pg. 6, lns. 19-20; pg. 7, ln. 2 - pg. 8, ln. 22; pg. 9, lns. 14-20; pg. 12, ln. 15 - pg. 13, ln. 18; pg. 15, lns. 8-18; pg. 19, lns. 14-24; pg. 24, lns. 8-12.

25 See at least App., FIGS. 1-3; pg. 2, lns. 31-33; pg. 4, lns. 16-18, 24-26; pg. 5, lns. 3-5 and 12-14; pg. 6, ln. 22 - pg. 7, ln. 11; pg. 8, lns. 11-22; pg. 9, lns. 7-20; pg. 11, lns. 6-14; pg. 13, lns. 12-22; pg. 23, lns. 17-24; pg. 24, lns. 8-14.

26 See for example, App., FIGS. 1-3; pg. 2, lns. 18-33; pg. 23, ln. 25 - pg. 24, ln. 14.

27 See at least App.; pg. 19, lns. 1-4.

28 See at least App., FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 10-15, 19-24 and 29 - pg. 5, ln. 1; pg. 5 lns. 6-10; pg. 6, ln. 24 - pg. 8, ln. 22; pg. 9, lns. 13-28; pg. 12, ln. 7 - pg. 13, ln. 30; pg. 15, ln. 8 - pg. 22, ln. 5; pg. 23, ln. 28 - pg. 24, ln. 8.

29 See at least App., FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 10-15, 22-24 and 29-30; pg. 5, lns. 2-5 and 10-12; pg. 7, lns. 5-11; pg. 7, ln. 22 - pg. 8, ln. 22; pg. 11, ln. 6 - pg. 12, ln. 6; pg. 13, lns. 17-28; pg. 15, lns. 1-7 and lns. 19-24; pg. 19, lns. 16-21; pg. 22, ln. 6 - pg. 23, ln. 16; pg. 24, lns. 1-10).

embodiments, code to generate an image as a function of the programmatic content and a representation of the audio/video content is also included in the computer program products where in some embodiments the representation of the audio/video content defines how the video/audio content is to be displayed (104, 108, 204, 210, 308, 310)³⁰. Further, in some embodiments, code is provided to format and combine the image with the video/audio content (110, 112, 212, 214, 312, 314).³¹

With regard to at least Claim 8, some embodiments provide systems that combine video/audio content (100, 102, 200, 202, 300, 302) with programmatic content (104, 208, 304). These systems include means for searching an instance of a source file for a variable and replacing the variable with a definition for the variable within that instance of the source file (100, 102, 106, 200, 202, 204, 300, 302).³² Further means for generating programmatic content as a result of the searching of the source files (104, 106, 204, 206, 304, 306) are provided.³³ Means can also be provided in these embodiments for generating an image as a function of the programmatic content and a representation of the audio/video content defining how the video/audio content is to be displayed (104, 108, 204, 210, 308, 310).³⁴ Additionally, means to combine the image with the video/audio content is included (110, 112, 212, 214, 312, 314).³⁵

Below is a table providing an association of the claimed means plus function structure and function of Claim 8 with the corresponding description in the specification as filed. This table may be similar to the summarization of the description above and examples of supporting description within the specification. The description above, however, is a

30 See for example App. FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 16-18 and 24-26; pg. 5, lns. 3-5 and 12-14; pg. 6, lns. 19-20; pg. 7, ln. 2 – pg. 8, ln. 22; pg. 9, lns. 14-20; pg. 12, ln. 15 – pg. 13, ln. 18; pg. 15, lns. 8-18; pg. 19, lns. 14-24; pg. 24, lns. 8-12.

31 See at least App., FIGS. 1-3; pg. 2, lns. 31-33; pg. 4, lns. 16-18, 24-26; pg. 5, lns. 3-5 and 12-14; pg. 6, ln. 22 – pg. 7, ln. 11; pg. 8, lns. 11-22; pg. 9, lns. 7-20; pg. 11, lns. 6-14; pg. 13, lns. 12-22; pg. 23, lns. 17-24; pg. 24, lns. 8-14.

32 See at least App., FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 10-15, 19-24 and 29 – pg. 5, ln. 1; pg. 5 lns. 6-10; pg. 6, ln. 24 – pg. 8, ln. 22; pg. 9, lns. 128; pg. 12, ln. 7 – pg. 13, ln. 30; pg. 15, ln. 8 – pg. 22, ln. 5; pg. 23, ln. 28 – pg. 24, ln. 8.

33 See at least App., FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 10-15, 22-24 and 29-30; pg. 5, lns. 2-5 and 10-12; pg. 7, lns. 5-11; pg. 7, ln. 22 – pg. 8, ln. 22; pg. 11, ln. 6 – pg. 12, ln. 6; pg. 13, lns. 17-28; pg. 15, lns. 1-7 and lns. 19-24; pg. 19, lns. 16-21; pg. 22, ln. 6 – pg. 23, ln. 16; pg. 24, lns. 1-10).

34 See for example App. FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 16-18 and 24-26; pg. 5, lns. 3-5 and 12-14; pg. 6, lns. 19-20; pg. 7, ln. 2 – pg. 8, ln. 22; pg. 9, lns. 14-20; pg. 12, ln. 15 – pg. 13, ln. 18; pg. 15, lns. 8-18; pg. 19, lns. 14-24; pg. 24, lns. 8-12.

35 See at least App., FIGS. 1-3; pg. 2, lns. 31-33; pg. 4, lns. 16-18, 24-26; pg. 5, lns. 3-5 and 12-14; pg. 6, ln. 22 – pg. 7, ln. 11; pg. 8, lns. 11-22; pg. 9, lns. 7-20; pg. 11, lns. 6-14; pg. 13, lns. 12-22; pg. 23, lns. 17-24; pg.

representative “summary” and the Appellants do not represent or intend that this table, or the accompanying references to the drawings and the specification, comprises an exhaustive presentation in this regard to the summaries above.

means for searching an instance of the source file for a variable, and for replacing the variable with a definition for the variable within that instance of the source file (100, 102, 106, 200, 202, 204, 300, 302)	See the application as filed, FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 10-15, 19-24 and 29 - pg. 5, ln. 1; pg. 5 lns. 6-10; pg. 6, ln. 24 - pg. 8, ln. 22; pg. 9, lns. 13-28; pg. 12, ln. 7 - pg. 13, ln. 30; pg. 15, ln. 8 - pg. 22, ln. 5; pg. 23, ln. 28 - pg. 24, ln. 8
means for generating programmatic content in response to the searching (104, 106, 204, 206, 304, 306)	See the application as filed, FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 10-15, 22-24 and 29-30; pg. 5, lns. 2-5 and 10-12; pg. 7, lns. 5-11; pg. 7, ln. 22 - pg. 8, ln. 22; pg. 11, ln. 6 - pg. 12, ln. 6; pg. 13, lns. 17-28; pg. 15, lns. 1-7 and lns. 19-24; pg. 19, lns. 16-21; pg. 22, ln. 6 - pg. 23, ln. 16; pg. 24, lns. 1-10
means for generating an image as a function of the programmatic content and a representation of the audio/video content, the representation of the audio/video content defining how the video/audio content is to be displayed (104, 108, 204, 210, 308, 310)	See the application as filed, FIGS. 1-3; pg. 2, lns. 27-33; pg. 4, lns. 16-18 and 24-26; pg. 5, lns. 3-5 and 12-14; pg. 6, lns. 19-20; pg. 7, ln. 2 - pg. 8, ln. 22; pg. 9, lns. 14-20; pg. 12, ln. 15 - pg. 13, ln. 18; pg. 15, lns. 8-18; pg. 19, lns. 14-24; pg. 24, lns. 8-12
means for combining the image with the video/audio content (110, 112, 212, 214, 312, 314)	See the application as filed, FIGS. 1-3; pg. 2, lns. 31-33; pg. 4, lns. 16-18, 24-26; pg. 5, lns. 3-5 and 12-14; pg. 6, ln. 22 - pg. 7, ln. 11; pg. 8, lns. 11-22; pg. 9, lns. 7-20; pg. 11, lns. 6-14; pg. 13, lns. 12-22; pg. 23, lns. 17-24; pg. 24, lns. 8-14

(6) Grounds of Rejection to be Reviewed on Appeal

Issue 1: whether claims 1-12 are anticipated under 35 U.S.C. § 102(b) by U.S. Patent No. 5,909,551 (Tahara, referred to below as the Tahara patent).

Issue 2: whether claim 9 is clear and has relationships between limitations.

Issue 3: whether Applicants’ prior arguments satisfied 37 C.F.R. §1.111(b).

(7) Arguments

Issue 1: claims 1-3, 23 and 25-26 are not anticipated under 35 U.S.C. § 102(b) by the Tahara patent.

Claim 1

Applicants respectfully submit that Tahara does not teach at least all of the limitations as recited in claim 1, and therefore, claim 1 is not anticipated by the Tahara patent. More specifically, claim 1 recited in part:

generating authoring output comprising a definition for a variable ...
selecting a source file, the source file comprising the variable;
searching an instance of the source file for the variable; and
replacing the variable with the definition for the variable within that instance of the source file such that the instance of the source file contains the definition for the variable...;
generating programmatic content in response to the searching.... (Claim 1).

First, Applicants respectfully submit that Tahara does not teach at least replacing the variable with the definition within the instance of the source file as recited in the pending claim 1. Instead, Tahara at best only describes identifying a link in a file and using that link to shift to another already existing file. In rejecting claim 1, the office action cites portions of col. 17, line 34 – col. 18, line 8 suggesting that Tahara describes a replacing of a variable within a source file with a definition such that the source file includes the definition. However, columns 17-18 do not teach or suggest the replacing as claimed in instead only describe shifting between files. Specifically, for example, Tahara states “when the user selects the selection button 2607 [1], shift is made to the corresponding page PAGE001.HTM file as illustrated ...” (Tahara, col. 17, lns. 65-67, emphasis added), and further in column 17 states “[w]hen the image is selected by the user and shift is to be made to any page ..., and the page name to be shifted to is designated ...” (Tahara, col. 17, lns. 46-49, emphasis added). Therefore, the Tahara patent does not teach or suggest replacing a variable in an instance of a source file with a definition for the variable such that the instance of the source file includes the definition.

The office action in rejection claim 1 suggests that Tahara describes replacing the variable with its definition stating that “Tahara discloses storing source files in a directory structure (Fig. 24) wherein specific browser type of files are evoked from a user choosing a button ...” (office action, page 4, emphasis added). As such, the office action admits that

multiple source files are stored on a CD 2301 of Fig. 23 in Tahara, and that one of these “source files” are accessed for use in displaying in response to “a user choosing a button.” The accessing and displaying of content of a subsequent file as described in Tahara cannot be equated to “replacing the variable with the definition for the variable within that instance of the source file such that the instance of the source file contains the definition for the variable” as recited in claim 1 (emphasis added). Accessing separate files cannot be equated to replacing within an instance of a source file a definition for a variable of the source file such that the instance of the source file include the definition. There is no replacing within the instance of the source file, and instead Tahara only describes “shifting” between files (e.g., see at least Tahara, col. 17-18).

The office action fails to show and Tahara fails to teach or suggest that a source file is altered such that a variable in an instance of a source file is replaced with a definition so that instance of the source file itself contains the definition. Tahara only describes accessing and shifting to already stored files in response to “a user choosing a button” to display content. When the hyperlink inside an initial markup file in Tahara is processed, it causes a shift to an alternate and existing file, and the variable in the initial file directs the shifting to the subsequent file (see for example, Tahara Figs. 27a, 27b, col. 17, lns. 44-67). Tahara does not suggest that the variable in the initial file is replaced such that the initial file includes the subsequent file (i.e., what the office action suggests is the definition for the variable). Instead, Tahara shows in Figs. 27a-b, cited in the office action, that a separate file (e.g., “PAGE001.HTM”) is accessed “when the user selects the selection button 2607 [1]” (Tahara, col. 17, lines 65-66). There is no discussion in Tahara that, for example, the variable “A HREF=” (as defined in the office action as being a variable) is replaced in the “INDEX.HTM” file with the “PAGE001.HTM” file as shown in Fig. 27b. Instead, Tahara specifically teaches away from such replacing in that Tahara describes shifting to the existing secondary file “PAGE001.HTM” to be accessed and displayed. Tahara further teaches away from altering the “INDEX.HTM” file in that Tahara specifically describes provisions to “return” to the “INDEX.HTM” (i.e., “RETURN” button, see for example, Tahara, col. 17, lines 15-16). Therefore, Tahara does not teach or suggest replacing the variable in the instance of the source file with the definition of the variable such that the instance of the

source file contains the definition for the variable as recited in claim 1. Thus, claim 1 is not anticipated by the Tahara patent.

Additionally, the office action in response to Applicants prior arguments filed June 28, 2006, actually supports Applicants' arguments stating that "Tahara discloses that another WWW page is created based on user selection; and since each instance of the page being displayed underlies its markup document, the event of fetching a corresponding definition inside each tagged data as explained in the Office Action is considered equivalent to replacing a source file ..." (office action, page 8, emphasis added). Therefore, the Examiner's contention supports Applicants arguments in that Tahara does not teach "replacing" a variable within an instance of a source file such that the source file contains the definition of the variable, and instead Tahara "replaces" or a source file with a subsequent existing source file, and at least does not teach or suggest replacing within a source file.

The office action continues in later arguments to further support Applicants' assertions stating "Tahara discloses browser processing based on the user triggering of HTML file in which hyperlinked variables ... are set for defining external data or other source files ... which are to be displayed..." (office action, page 4, emphasis added). Again, the Tahara patent describes linking to other existing files to display separate content, and does not teach or suggest replacing a variable such that an instance of the source file contains the definition for the variable.

Further, the Tahara patent fails to describe or suggest replacing the variable with the definition for the variable within that instance of the source file. The office action suggests that the Tahara patent describes replacing the variable with the definition for the variable where "variable enclosed and defined within markup tags and being replaced by corresponding html page/file or image data reads on replacing variable <.../> with its definition being enclosed" (office action, pg. 3). The office action equates the browser type files stored in the WWW directory with the source file recited in claim 1 (see office action, pg. 3-4). However, when the hyperlink inside the markup file in Tahara is processed, it causes a shift to an alternate and existing page, and the variable is not replaced with the definition within the markup file such that the instance of the markup file contains the subsequent file (the definition for the variable) (see, Figs. 27 a, 27 b, col. 17, lns. 44-67). The variable within the source file is never changed and remains the same, the selection of a

button by a user only causes shifting to a new file and does not cause any replacing of the variable within the instance of the source file where the variable is located with the definition of the variable. For example, according to Tahara if the user selects to return to a first page the same variable will be in the first page as it was before the shifting to another page occurred, therefore, the definition of the variable is not replaced (see at least Tahara, col. 17, ln. 41 – col. 18, ln. 45). The Tahara patent does not teach replacing the variable as claimed, but instead only describes shifting from one existing file to another predefined and stored page file on processing the user selection (Tahara, col. 17, lns. 65-67). The variable is not changed within either of the files. Instead, the browser is redirected to a different already stored file based on user selections within a displayed page (see, for example, Figs. 27a and 27b). Therefore, the Tahara patent does not teach at least the “replacing the variable with the definition for the variable within that instance of the source file such that the instance of the source file contains the definition for the variable” as recited in claim 1, and thus, claim 1 is not anticipated by Tahara.

The office action on page 4 further suggests “Tahara discloses a source file being retrieved - - via search directory - - wherein a tagged variable is automatically processed and replaced inside the file or page (e.g., col. 17, line 34 to col. 18, line 3; Fig. 24-26; col. 17, line 53 to col. 18, line 8, i.e. tagged variables being replaced inside the same page).” However, claim 1 does not recite replacing a variable inside a displayed page, but instead recites replacing a variable in an instance of a source file such that the instance of the source file includes the definition for the variable. The source file cannot be equated to the displayed content. The source files may dictate what is displayed, but the source files themselves are not altered and Tahara does not teach or suggest that these source files are altered to include other files. The Tahara patent specifically distinguishes the source files from the displayed content in that it describes shifting between files to alter what is displayed. The files are not altered, just what is displayed. For example, Tahara specifically distinguishes between the files and what is displayed stating “[i]nteractive image reproducing means 2302 reproduces data to the user fro the interactive image recording medium 2301” (Tahara, col. 15, lns. 46-48, emphasis added). Tahara continues stating, for example, that “[i]n FIG. 27a the file INDEX.HTM is opened, the image file MENU001.GIF is displayed and then the image file KEY1.GIF is displayed” (Tahara, col. 17, lines 50-52, emphasis added) distinguishing

between the recorded files and the display of those content of those files where the file is used to generate the display but the display is not the file. Therefore, altering a display cannot be equated to altering the source file as Tahara specifically teaches away from altering the source files.

Further, the rationale of the office action is inconsistent. The office action initially states that a definition within a file is already defined in that file. Specifically, the office action states that “variable enclosed and defined within markup tags and being replaced by corresponding html page/file or image data reads on replacing the variable <...> with its definition” (office action, page 5, emphasis added). Therefore, the office action admits that the variable is already defined within the file. As such, Tahara does not teach replacing “the variable within that instance of the source file” as claimed, and instead Tahara describes using the definition to link to another already existing file. Nowhere does Tahara suggest that the instance of the file is altered such that the variable is replaced in that instance of the source file with a definition.

Second, Applicants respectfully submit that the Tahara patent fails to teach or suggest at least “generating programmatic content in response to the searching” as recited in claim 1. The office action suggests that Tahara describes “hyperlinked variables ... [that] are set for defining external data or other source files” (office action, page 4). However, a variable that directs a system to existing content already stored on the CD 2301 cannot be equated to generating content in response to the searching for a variable. Instead, Tahara already has the subsequent content or files created and stored. Based on the logic of the office action, the user’s selection of the hyperlink causes the searching. However, the hyperlink is stored on the CD with the related content or file associated with the hyperlink. Therefore, the content already exists, and thus, is not generated in response to the searching. Instead, the existing content associated with the link is simply accessed and potentially displayed, but not generated in response to searching. For example, Fig. 24 of Tahara shows part of the content of the CD 2301 depicted in Fig. 23 with the content already incorporated on to the CD to allow the shifting between files.

The office action in rejection claim 1 cites to “HTML 2306; PC 2307 – Fig. 23” (office action, page 5) in suggesting that Tahara describes generating programmatic. However, as described by Tahara, and stated by the office action, when the markup file is

selected by “the user choosing a button ... Tahara discloses a source file being retrieved” (office action, page 4, emphasis added). Therefore, the office action clearly states that the source file associated with the hyperlink already exists and is “retrieved.” There is no suggestion according to Tahara that the content is generated in response to the user selection. According to Tahara, the selection of a button causes the access to an existing file (e.g., Tahara, col. 17, lns. 45-52). For example, Tahara further states that:

when the user selects the selection button [3], shift is made to the corresponding page PAGE003.HTM file ... In PAGE003.HTM of FIG. 28a, a URL address is defined as a page shifted to on selection of the character 2608 ‘Order: \$200’ by the user. (Tahara, col. 18, lines 46-52, emphasis added).

The file PAGE003.HTM stored on the CD is retrieved. The data in the HTML files remain the same after the shift to an alternate and existing file occurs, and no programmatic content is generated or written into the HTML 2306 or PC 2307 (Tahara, col. 17, ln. 65 – col. 18, ln. 27). Further, the HTML file 2306 that the office action attempts to equate to the claimed programmatic content is predefined and stored on the medium 2301. Therefore, it is not generated in response to the searching for the hyperlink.

Furthermore, the office action on page 8 states that “[t]he action of selecting an iconic representation of a WWW file stored inside a directory (see Fig. 24) by Tahara, teaching that a file is being searched ... in order to have it displayed” (emphasis added). Therefore, the office action supports Applicants’ assertion that the Tahara patent does not describe generating programmatic content in response to searching, but instead accesses already stored files. Therefore, the HTML file 2306 is not generated in response to searching but instead is previously stored and accessed. Tahara does not teach or suggest generating programmatic content in response to searching, and instead only describes a user selection causing a transition to existing and stored content (see at least, Tahara, col. 16, lns. 31-47, and col. 17, lns. 65-67). As such, the Tahara patent does not teach or suggest at least generating programmatic content in response to the searching as claimed.

Third, the office action suggests that the Tahara patent describes selecting a source file, the source file comprising the variable and searching an instance of the source file for the variable where “scanning through the directory of WWW data of the hypertext content stored ... will read on search for an instance of a source file containing a variable” (Office Action, pg. 3, emphasis added). However, claim 1 does not recite searching for an instance of a

source file containing a variable as suggested by the office action. Instead, claim 1 recites selecting a source file and “search an instance of the source file for the variable” (emphasis added). Scanning through the directory of WWW data does not teach searching through an instance of a source file for a variable, and instead only describes searching for a source file. The office action further argues that “[t]he action of selecting an iconic representation of a WWW file stored inside a directory ... by Tahara, teaching that a file is being searched in this tree in order to have it displayed, would be equivalent to searching for an instance of a file” (office action, pg. 8, emphasis added). Again, claim 1 provides for the searching of an instance of a source file for the variable, and not searching a directory for a source file as suggested in the office action. Searching for a file cannot be equated to the searching an instance of a file for a variable as suggested in the office action.

Finally with regard to at least claim 1, the office action suggests in response to Applicants’ prior arguments that “[t]he claim lacks clear implementation details” (office action, page 8). However, the claim does not lack details in that the claim specifically recites “replacing the variable with the definition for the variable within that instance of the source file such that the instance of the source file contains the definition for the variable” (claim 1). The replacing of the variable with the definition is not a general replacement, but instead is a replacement into the source file so that the source file contains the definition. The Tahara patent does not teach or suggest replacing within a source file a definition for a variable such that the source file contains the definition. Therefore, the Tahara patent fails to teach each limitation as recited in at least claim 1.

As such, the Tahara patent fails to describe or suggest all of the limitations as recited in claim 1. Therefore, claim 1 is in condition for allowance.

Claim 8

Similarly, the Tahara patent fails to teach each limitation as recited in claim 8. More specifically, for example, the Tahara patent fails to teach at least “means for searching an instance of the source file for a variable, and for replacing the variable with a definition for the variable within that instance of the source file” as recited in claim 8.

The office action suggests that the Tahara patent describes selecting a source file, the source file comprising the variable and searching an instance of the source file “in light of the

rationale as set forth in claim 1” (office action, pg. 5). In rejecting claim 1 the office actions suggests that Tahara describes selecting a source file, the source file comprising the variable by stating that “scanning through the directory of WWW data of the hypertext content stored ... will read on search for an instance of a source file containing a variable” (Office Action, pg. 3, emphasis added). The office action continues to state that “[t]he action of selecting an iconic representation of a WWW file stored inside a directory ... by Tahara, teaching that a file is being searched in this tree in order to have it displayed, would be equivalent to searching for an instance of a file” (office action, pg. 8). However, claim 8 does not recite searching for an instance of a source file containing a variable as suggested by the office action and instead recites searching an instance of the source file for a variable. Scanning through the directory of WWW data does not teach searching through an instance of a source file selected for a variable, and instead describes searching for a source file.

Further, the rational of the office action is inconsistent. The office action initially states that a definition within a file is already defined in that file. Specifically, the office action states that “variable enclosed and defined within markup tags and being replaced by corresponding html page/file or image data reads on replacing the variable <...> with its definition” (office action, page 5, emphasis added). Therefore, the office action admits that the variable is already defined within the file. As such, Tahara does not teach replacing “the variable within that instance of the source file” as claimed, and instead Tahara describes using the definition to link to another already existing file. Nowhere does Tahara suggest that the instance of the file is altered such that the variable is replaced in that instance of the source file with a definition.

Furthermore, the Tahara patent fails to describe or suggest replacing the variable with the definition for the variable within that instance of the source file. The office action at page 5 suggests that the Tahara patent describes replacing the variable with the definition for the variable where the “variable enclosed and defined within markup tags and being replaced by corresponding html page/file or image data reads on replacing variable <.../> with its definition being enclosed.” The office action equates the browser type files of Tahara stored in the WWW directory with the source file recited in claim 8 (see Office Action, pg. 3-4).

According to Tahara, however, when the hyperlinked definition inside the markup file is processed, it shifts to an alternate and existing file within the WWW directory, and the

definition of the variable is not changed within the markup file such that the instance of the markup file contains file (see, at least col. 17, lns. 44-67). The variable is never replaced within a source file, and further the definition of the variable within the source file does not change and instead remains the same. According to Tahara, the definition for the “variable” as defined by the office action only causes shifting to a new file to display a new page (see at least Tahara, col. 16, lns. 34-47 and col. 17, lns. 43-52) and does not cause any replacing of the variable definition within the instance of the source file where the variable is located. For example, Tahara specifically states that “[w]hen the image is selected by the user and a shift is to be made to any page ...and the page name to be shifted to is designated by the portion ‘...’” (col. 17, lns. 46-50, emphasis added), and continues stating “[i]n the display window 2601, when the user selects the select button 2607 [1], shift is made to the corresponding page PAGE001.HTM file...” (col. 17, lns. 65-67, emphasis added). As a further example, according to Tahara, if the user selects to return to the first page the same variable is still within the first page as it was before the searching occurred (e.g., see at least Tahara, col. 17, lns. 15-16), therefore, the variable is not replaced.

The Tahara patent does not teach at least replacing the variable as claimed, and instead only describes shifting from one existing file to another predefined and stored file based on processing the user selection (e.g., see at least Tahara, col. 17, lns. 65-67). The definition of a variable is not changed within either of the files. Instead, the browser is redirected to a different already stored file based on user selections within a displayed page (Figs. 27a and 27b). Therefore, Tahara does not teach or suggest replacing the variable with a definition for the variable within that instance of the source file.

The Office Action further suggests that “since each instance of the page being displayed underlies its markup document, the event of fetching a corresponding definition inside each tagged data as explained in the Office Action is considered equivalent to replacing a source file being selected via the GUI wherein a variable inside such file is being replaced by its definition for the user to see its defined content” (Office Action, pg. 8). Applicants respectfully submit that the Examiner’s logic supports Applicants arguments in that the office action admits that Tahara shifts from one file another file, and Tahara does not teach replacing that variable with the definition for the variable within the source file.

Further, this argument presented in the office action fails to show where Tahara teaches replacing the variable with the definition for the variable within an instance of the source file.

Still further, the Tahara patent fails to describe or suggest means for generating programmatic content in response to the searching as recited in claim 8. The Office Action equates HTML 2306 and PC 2307 described in Tahara with the programmatic content recited in claim 8. However, the contents of the HTML 2306 structure contain the files that the Office Action equates with the source file being searched and replaced (Office Action, pg. 5). The programmatic content recited in claim 8 is dependent on the searching of the source file recited in claim 8 and the generating of programmatic content occurs in response to the searching of the source file. As such, the HTML 2306 and PC 2307 cannot be equated with the programmatic content because equating these elements with the programmatic content recited disregard the dependency of the limitations recited in claim 8, and more specifically, the dependency between searching a source file and generating programmatic content.

Additionally, the system described in the Tahara patent does not generate programmatic content, nor does Tahara describe generating HTML data as a result of searching or even the browser scanning through the hierarchy and linking to other already existing data. Instead, as described by Tahara, and stated by the office action, when the markup file is selected by the user choosing a button and after the search occurs, the value for the button is used to link to other data files that define displays for viewing purposes (see for example, Tahara, col. 18, lns. 14-16, and office action, pg. 5). Further, the office action states that "Tahara discloses browser processing based on the user triggering of HTML file in which hyperlinked variables (e.g. col. 17, lines 56-64; Fig. 27) are set for defining external data or other source files (col. 18, lines 23-27, 46-52) which are to be displayed following the interactive selection by the user" (office action, page 4). Therefore, as a result of the selection of the file no new programmatic content is generated, and instead previously defined and stored files are accessed for displaying (e.g., see at least Tahara, col. 17, lns. 45-52 and col. 18, lns. 14-16). The data in the HTML files remain the same after the shift to an alternate file has been accessed and no programmatic content is generated or written into the HTML or PC data files (Tahara, col. 17, lns. 65-67 and col. 18, lns. 1-27).

Further, the end result of the process of accessing an alternate file associated with the HREF tag as described in the Tahara patent does not generate programmatic content, but

instead shifts to an already stored file that is displayed (see at least Tahara, Fig. 26, 27a-b; col. 17, lns. 61-64). As such, the Tahara patent fails to teach generating programmatic content in response to the searching as recited in claim 8.

Still further, in response to Applicants' previous arguments that the generating of programmatic content cannot be equated with the shifting of already existing HTML data, the office action states that the "[w]hat is recited as programmatic content has been perceived as something being result of a searching," and therefore, the Office Action asserts that the programmatic content can be equated to the same source files displayed and searched. However, the office action fails to consider every limitation as recited in the claim. More specifically, claim 8 recites generating programmatic content in response to the searching, not just searching. Therefore, the fact that Tahara may describe searching for a file cannot be equated to the generation of programmatic content in response to the searching. Additionally, searching source files and referring to the same source files as programmatic content does not read on generating programmatic content and instead only suggests shifting to existing content. As such, at least claim 8 is not anticipated by the Tahara patent, and hence, is in condition for allowance.

Claim 9

At least with respect to claim 9, the Tahara patent fails to teach each limitation as recited, and therefore, the Tahara patent does fails to anticipate claim 9 as recited. For example, the Tahara patent fails to teach at least "code to parse a source file searching for a variable, replacing the variable with the definition for the variable" as recited in claim 9.

More specifically, the Tahara patent fails to describe or suggest replacing the variable with the definition for the variable within that instance of the source file. In suggesting that the Tahara patent teaches this limitation recited in claim 9 the office action relies on the rationale in Claim 1 (office action, pg. 6). In rejecting claim 1, the office action suggests that the Tahara patent describes replacing the variable with the definition for the variable where "variable enclosed and defined within markup tags and being replaced by corresponding html page/file or image data reads on replacing variable <.../> with its definition being enclosed" (office action, pg. 3). As such, the office action equates the browser type files stored in the WWW directory with the source file recited in claim 1. However, when the hyperlinked

definition inside the markup file in Tahara is processed, it shifts to an alternate and existing page, and the definition of the variable is not replaced within the markup file (see, Figs. 27 a, 27 b, col. 17, lns. 44-67). The definition of the variable within the source file is never replaced or changed and remains the same, the variable only causes shifting to an existing file and does not cause any replacing of the variable with a definition within the instance of the source file. For example if the user selects to return to the first page the same variable will be within the file associated with the displayed first page as was before the searching occurred, therefore, the variable is not replaced with the definition within the source file. The Tahara patent does not teach replacing the variable as claimed, but instead only describes shifting from one file to another predefined and stored file based on processing the user selection (see at least Tahara, col. 17, lns. 65-67). The definition of a variable is not changed within either of the files. Instead, the browser is redirected to a different already stored file based on user selections (Figs. 27a and 27b), and the file is not altered such that a variable is replaced with a definition within that file.

Further, the Tahara patent fails to describe or suggest at least “generating programmatic content in response to the searching” as recited in claim 9. The office action equates HTML 2306 and PC 2307 described in Tahara with the programmatic content recited in claim 9. However, the office action also equates this content with the authoring output (Office Action, pg. 3). Further, the contents of the HTML 2306 structure contain the source files that the office action equates with the source file being searched (Office Action, pg. 3). However, the programmatic content recited in claim 9 is dependent on the searching of the source file recited in claim 9 and the generating of programmatic content occurs in response to the searching of the source file, as recited in claim 9. As such, the HTML 2306 and PC 2307 cannot be equated with the programmatic content, because they are equated with another element recited in claim 9, and further, equating these elements with the programmatic content recited in claim 9 disregard the dependency of the limitations recited in claim 9. Claim 9 does not state searching files. Instead, claim 9 provides for generating programmatic content in response to searching a source file for a variable.

Still further, in response to Applicants’ previous arguments that the generating of programmatic content cannot be equated with the shifting to already existing HTML data, the office action states that “[w]hat is recited as programmatic content has been perceived as

something being result of a searching,” and therefore, the office action asserts that the programmatic content can be equated to the same source files displayed and searched. However, the office action fails to consider every limitation as recited in the claim. More specifically, claim 9 recites generating programmatic content in response to the searching, referring to the same source files as programmatic content does not read on generating content and instead only suggests referencing existing content. As such, the Tahara patent fails to describe or suggest all of the limitations as recited in claim 9, and therefore, claim 9 is not anticipated by the Tahara patent.

Claim 10

With regard to at least claim 10, Applicants respectfully submit that Tahara does not teach at least all of the limitations as recited, and therefore, claim 10 is also not anticipated by the Tahara patent. More specifically, the Tahara patent does not teach at least “code to select a source file comprising at least one variable and to create a copy of the source file” as recited in claim 10. In rejecting claim 10, the office action relies on the same rationale as previously stated with respect to the rejection of claim 1 (office action, pg. 7). However, nowhere was it discussed with respect to claim 1 at least the creating of “a copy of the source file.” Further, the office action fails to even acknowledge that claim 10 includes limitations not recited in claim 1, and fails to address, for example, that claim 10 recite “code ... to create a copy of the source file” as recited in claim 10. Therefore, the office action has failed to show how the Tahara patent teaches each limitation of claim 1, and thus, has failed to establish a *prima facie* case of anticipation.

Further, Applicants respectfully submit that the Tahara patent does not teach or suggest at least “code ... to create a copy of the source file” (claim 10). There is no discussion or suggestion in Tahara to generate a copy of a source file. Still further, there is no discussion or suggestion in Tahara to at least “code ... to replace the variable with a definition for the variable within the copy of the source file” as recited in claim 10. Therefore, claim 10 is not anticipated by Tahara.

Still further, the Tahara patent fails to describe or suggest at least “code ... to replace the variable with a definition for the variable within the copy of the source file” as recited in claim 10. The office action suggests that the Tahara patent describes replacing the variable

with the definition for the variable where “variable enclosed and defined within markup tags and being replaced by corresponding html page/file or image data reads on replacing variable <.../> with its definition being enclosed” (Office Action, pg. 3). The office action equates the browser type files stored in the WWW directory with the copy of the source file recited in claim 10 (see office action, pg. 3-4). However, when the hyperlinked definition inside the markup file in Tahara is processed, it shifts to an alternate and existing file, and the definition of the variable is not changed within the markup file such that the instance of the markup file contains the definition for the variable (see, Figs. 27 a, 27 b, col. 17, lns. 44-67). The definition of the variable within the source file is never changed and remains the same, selection by the user only causes shifting to a new file and does not cause any replacing of the variable definition within the instance of the source file, nor the replacing of the variable with a definition for the variable within the copy of the source file. For example, as described in Tahara, when a user selects to return to a previous page the same variable will be in the source file used to display the first page. Therefore, the variable is not replaced by a definition for the variable, and further, a variable is not replaced by a definition within a copy of the source file. The Tahara patent does not teach at least replacing the variable as claimed, but instead only describes shifting from one file to another predefined and stored file based on processing the user selection (Tahara, col. 17, lns. 65-67). The variable is not changed within either of the files. Instead, the browser is redirected to a different already stored file based on user selections within a page (Figs. 27a and 27b).

Additionally, the Tahara patent fails to describe or suggest at least “code to generate programmatic content in response to the searching” as recited in claim 10. The Office Action equates HTML 2306 and PC 2307 described in Tahara with the programmatic content recited in claim 1 (Office Action, pg. 3). The HTML 2306 and PC 2307, however, are not generated in response to the searching for the variable within the source file. Instead, the HTML 2306 and PC 2307 are previously generated and stored. There is no generation of the HTML or PC in response to searching. The searching arguably just accesses other files in the HTML 2306 or PC 2307. The Tahara patent fails to suggest that HTML 2306 or PC 2307 is generated in response to the searching as recited in claim 10. Therefore, the Tahara patent does not teach each limitation as claimed and claim 10 is not anticipated by Tahara.

Furthermore, the office action in attempting to support the rejection of claim 10 continues to state that:

processing based on the user triggering of HTML file in which hyperlinked variables ... are set for defining external data or other source files ... which are to be displayed following the interactive selection by user, and resolving of hyperlinked variables - - from that browser in response to that selection - - into displayed content to be finally stored or recorded into the different section of image medium 2301...
(office action, pgs. 4-5).

The Tahara patent, however, does not teach or suggest processing based on a “user triggering” that content is to be “finally stored or recorded” onto the media image 2301 as suggested in the office action. Tahara, instead teaches away from such operation in that the image medium 2301 is generated before any “user triggering”, and in fact cannot operate as described without the content being stored on the CD 2301. Therefore, this logic is inconsistent with the limitations of claim 10, and further, Applicants respectfully submit that Tahara does not teach and instead teaches away from generating programmatic content in response the searching. Thus, claim 10 is not anticipated by Tahara.

Additionally, Tahara describes that when the markup file is selected by the user choosing a button, the value for the button is used to shift to another data file for viewing purposes (e.g., see at least Tahara, col. 17, lns. 45-52). Therefore, as a result of the selection of a button no new programmatic content is generated, and instead a previously defined file is accessed and used in displaying (e.g., Tahara, col. 17, lns. 45-52). The data in the HTML 2306 and PC 2307 remain the same after the shift to an alternate file has occurred and no programmatic content is generated or written to the medium 2301 (Tahara, col. 17, ln. 65 - col. 18, ln. 27). Further, the HTML file 2306 that the office action attempts to equate to the claimed programmatic content is predefined and stored on the CD 2301. Therefore, the HTML file 2306 is not generated in response to searching and no changes are made to this HTML file after the user selection causes a transition to the alternate file (e.g., Tahara, col. 16, lns. 31-47, and col. 17, lns. 65-67). As such, the Tahara patent does not teach or suggest at least generating programmatic content in response to the searching as claimed, and thus, claim 10 is not anticipated by Tahara.

Claim 11

Applicants respectfully submit that the Tahara patent fails to teach or suggest at least “the instance of the source file is a copy of the source file.” The office action in rejection claim 11 cites to claim 10 and Figs. 2 and 24 of Tahara. However, claim 10 fails to suggest anything about a copy of a source file. Instead, claim 10 only describes “image reproduction control data” to facilitate the visual reproduction or depiction of that defined by first image data. There is no suggestion or teach that in instance of a source file is a copy of a source file.

Further, Figs. 2 and 24 of Tahara fail to suggest there are copies of source file. These figures only show directory structures with files. There is no suggestion of a copy of a source file.

Still further, claim 11 does not simply provide for a copy of a source file, but instead provides that an instance of a source file that is a copy of a source file is searched for a variable and where that variable is replaced by a definition of the variable such that the instance of the source file includes the definition. Tahara fails to teach or suggest such a copy of a source file. Therefore, claim 11 is also not anticipated by Tahara.

Claim 12

Applicants respectfully submit that the Tahara patent fails to teach or suggest at least “the instance of the source file is a copy of the source file.” The office action in rejection claim 12 cites to claim 10 and Figs. 2 and 24 of Tahara. However, claim 10 fails to suggest anything about a copy of a source file. Instead, claim 10 only describes “image reproduction control data” to facilitate the visual reproduction or depiction of that defined by first image data. There is no suggestion or teach that in instance of a source file is a copy of a source file.

Further, Figs. 2 and 24 of Tahara fail to suggest there are copies of source file. These figures only show directory structures with files. There is no suggestion of a copy of a source file.

Still further, claim 12 does not simply provide for a copy of a source file, but instead provides that an instance of a source file that is a copy of a source file is searched for a

variable and where that variable is replaced by a definition of the variable. Tahara fails to teach or suggest such a copy of a source file. Therefore, claim 12 is also not anticipated by Tahara.

Issue 2: Claim 9 does not need to be amended as claim 9 already provides semantic and timing relationships between elements of the claim.

The final office action suggests that claim 9 fails to “put forth the semantic and/or timing relationship between the limitation of ‘to parse’ and” other identified claim elements (office action, pg. 2). Applicants respectfully submit, however, that claim 9 does not need to be amended as the claim already provides for semantic and timing relationships. Specifically, the claim recites code to parse that provides “searching for a variable” and “replacing the variable...” Applicants submit that the code cannot provide replacing of the variable until the variable is search for and found. Therefore, the claim includes timing. Further, the claim recites “generating programmatic content in response to searching.” Again, the claim already includes timing in that the “generating programmatic content” is preformed in response to the “searching,” and thus, does not occur prior to “searching.” Claim 9 particularly points out and distinctly claims the subject matter that would be fully understood by one skilled in the art at least in view of the specification and further provides interrelationships between claim limitations. Therefore, the claim already provides semantic and timing relationships between claim elements, and thus, claim 9 does not need to be amended and Applicants respectfully request the objection be withdrawn.

Issue 3: Applicants’ prior arguments satisfied 37 C.F.R. §1.111(b).

The Examiner argued in response to Applicants’ previous amendment, that Applicants’ arguments “fail to comply with 37 C.F.R. §1.111(b)” (office action, pgs. 8-9). In support of this assertion, the Examiner suggests that

The claim lacks clear implementation details ..., nor does it not impart appropriate relationship between elements recited; nor does it give clear context of cause-to-effect ... That is: where does this (variable) definition come from and how is it implemented as? Where does this one instance of source file come from and what does it amount to? Where does the replacement take place and where does the replacement instance of file end up in the context of programmatic content ... How is search as recited related to replacement of definition ... How does one instance having its variable

resolved affect the search or the crating of image? What makes the final creation a useful image (Office action, pgs. 8-9)

Applicants respectfully traverse these assertions. The claim is clear and would be fully understood by one skilled in the art. Further, the Applicants' arguments distinguished the claims over the cited art and specifically referenced the Tahara patent countering the Examiner's assertions, demonstrating the distinctions and showing where the Tahara patent failed to teach all of the claimed limitations.

Additionally, the office action fails to take into consideration that the claims clearly provide relationships between elements. Instead, the claims extensively provide interrelationships between elements. For example, the claim provides interrelationships between claim elements, such as the searching of an instance of a source file for the variable of the source file, replacing within that instance of the source file so that instance contains the definition for the variable, generating programmatic content in response to the searching, generating an image as a function of the programmatic content, and other interdependencies. Applicants respectfully submit that the Examiner is requesting the additional limitations be incorporated that are not needed as the claim is clear and one skilled in the art would fully understand the claim as recited at least in view of the specification.

After suggesting that the claims fail to provide interrelationships, the Examiner suggests that Applicants arguments fail to comply with 37 C.F.R. §1.111(b) "because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references." Applicants respectfully traverse this assertion. Applications presented extensive arguments and examples of how the claims distinguish from the claims, and how Applicants believed the assertions presented by the Examiner were incorrect and where the Examiner's assertions did not have logical support. Therefore, Applicants' respectfully that the prior response complied with 37 C.F.R. §1.111(b) and request this assertion be withdrawn.

(8) Claim Appendix

Claim 1: A method for combining video/audio content with programmatic content comprising:

generating authoring output comprising a definition for a variable, and further comprising a representation of the video/audio content, the representation of the video/audio content defining how the video/audio content is to be displayed;

selecting a source file, the source file comprising the variable;

searching an instance of the source file for the variable, and replacing the variable with the definition for the variable within that instance of the source file such that the instance of the source file contains the definition for the variable;

generating programmatic content in response to the searching;

generating an image as a function of the programmatic content and the representation of the video/audio content; and

combining the image with the video/audio content.

Claim 2: The method of Claim 1 further comprising storing said image to a storage medium.

Claim 3: The method of Claim 1 further comprising transmitting said image through a transmission medium.

Claim 4: The method of Claim 1 wherein said searching includes searching said source file at build time.

Claim 5: The method of Claim 1 wherein said searching includes searching said source file at run time.

Claim 6: The method of Claim 5 wherein said searching includes searching in response to a software engine executed on a browser.

Claim 7: The method of Claim 5 wherein said searching includes searching in response to the insertion of a DVD into a hardware device.

Claim 8: A system for combining video/audio content with programmatic content comprising:

- means for searching an instance of the source file for a variable, and for replacing the variable with a definition for the variable within that instance of the source file;
- means for generating programmatic content in response to the searching;
- means for generating an image as a function of the programmatic content and a representation of the audio/video content, the representation of the audio/video content defining how the video/audio content is to be displayed; and
- means for combining the image with the video/audio content.

Claim 9: A computer program product stored on a computer-readable medium for use in combining video/audio content with programmatic content comprising:

- code to parse a source file searching for a variable, replacing the variable with a definition for the variable within the source file, and generating programmatic content in response to the searching;
- code to generate an image as a function of the programmatic content and a representation of the audio/video content, the representation of the audio/video content defining how the video/audio content is to be displayed; and
- code to format and combine the image with the video/audio content.

Claim 10: A software system stored on a computer readable medium comprising:

- code to select a source file comprising at least one variable and to create a copy of the source file;
- code to search the source file for the variable, and to replace the variable with a definition for the variable within the copy of the source file;
- code to generate programmatic content in response to the searching;

code to generate an image as a function of the programmatic content and a representation of the audio/video content, the representation of the audio/video content defining how video/audio content is to be displayed; and
code to format and combine the image with the video/audio content.

Claim 11: The method of claim 1, wherein the instance of the source file is a copy of the source file.

Claim 12: The system of claim 8, wherein the instance of the source file is a copy of the source file.

(9) Evidence Appendix

None.

(10) Related Proceedings Appendix

None.

CONCLUSION

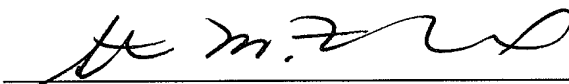
Appellants submit that the rejections of the pending claims 1-12 are in err, and that claims 1-12 are patentable over the applied reference.

Appellants respectfully request a reversal of the final rejection.

Dated:

9/7/07

Respectfully submitted,



Steven M. Freeland
Registration No. 42,555

Address all correspondence to:

FITCH, EVEN, TABIN & FLANNERY
Thomas F. Lebens
120 South LaSalle, Ste. 1600
Chicago, Illinois 60603-3406
(858) 552-1311